

B. AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1-39 (Cancelled).

Claim 40 (Currently Amended): A computer implemented method for replicating a software application, the computer implemented method comprising:

- executing the software application on a primary node to form a master application;
- identifying resources and dependencies required by the master application to form required resources;
- updating the required resources dynamically on the primary node;
- generating a structure of the master application and a dynamic graph of the required resources from the required resources;
- replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica by checkpointing the resources on the set of secondary nodes [[:]], wherein the set of secondary nodes comprises one or more secondary nodes, wherein the replicated resources on the set of secondary nodes comprise a virtual memory and calling stack of each process of the master application, at least one system resource used by the master application, and data written on disks used by the master application, wherein the checkpointing has an adjustable period to optimize the difference between the recovery time after switching to the set of secondary nodes and the quantity of information to be captured and transferred to the set of secondary nodes;
- restoring the replica on the set of secondary nodes to form a set of clone software applications, wherein the set of clone software applications comprises one or more clone software applications;
- executing the set of clone software applications on the set of secondary nodes, without loss of context; and

updating the set of clone software applications with incremental updates of the required resources of the master application to create a hot standby application.

Claim 41 (Previously Presented): The computer implemented method according to claim 40, wherein replicating the resources further comprises:
creating and maintaining a dependency tree, based on the dynamic graph, supplying, at all times, information on the replicated resources.

Claim 42 (Canceled).

Claim 43 (Currently Amended): The computer implemented method according to claim 4[[2]]0, wherein replicating the resources further comprises:

capturing the resources on the primary node to create captured required resources;

transferring the captured required resources over the network to the set of secondary nodes; and

restoring the captured required resources on the set of secondary nodes.

Claim 44 (Currently Amended): The computer implemented method according to claim 4[[2]]0, wherein replicating the resources further comprises:

optimizing the checkpointing.

Claim 45 (Previously Presented): The computer implemented method according to claim 44, wherein the checkpointing is incremental.

Claim 46 (Previously Presented): The computer implemented method according to claim 44, wherein the checkpointing is discriminating.

Claim 47 (Currently Amended): The computer implemented method according to claim 4[[2]]0, wherein the checkpointing further comprises at least one of the following:

- processing a synchronization barrier for locking the processes of the master application to take a non-blurred photograph of the state of the primary node and the master application at each of a plurality of phases of the checkpointing;

- managing resources necessary for replicating the master application during the plurality of phases of the checkpointing;

- managing the at least one system resource used by the master application during the plurality of phases of the checkpointing; and

- managing process resources.

Claim 48 (Previously Presented): The computer implemented method according to claim 40, wherein replicating the resources further comprises:

- replicating applicative data files between the primary node, whereon the software application is run, and a stand-by node.

Claim 49 (Previously Presented): The computer implemented method according to claim 40, wherein replicating the resources further comprises:

- ensuring functional continuity of the software application in a multi-computer architecture cluster, the software application being executed at a given time on one of the computers of the cluster, called the primary node, while other computers of the cluster are called a set of secondary nodes, wherein ensuring functional continuity further comprises:

- replicating the software application on at least one of the secondary nodes to provide a set of clones of the application, wherein the set of clones comprises one or more clones;

- updating the set of clones, and

- responsive to detecting an event affecting the primary node, switching from the software application being executed on the primary node, to the software application being executed on the set of clones.

Claim 50 (Previously Presented): The computer implemented method according to claim 49, wherein replicating the software application is of a holistic nature.

Claim 51 (Previously Presented): The computer implemented method according to claim 49, wherein updating the set of clones further comprises updating the set of clones of the application.

Claim 52 (Previously Presented): The computer implemented method according to claim 49, wherein ensuring functional continuity further comprises supervising a state of the resources necessary to operate the software application.

Claim 53 (Previously Presented): The computer implemented method according to claim 49, wherein detecting an event affecting the primary node further comprises:
responsive to detecting an event affecting the primary node, electing a clone to be substituted for the primary node of the software application, wherein the secondary node on which the clone elect is installed becomes a new primary node.

Claim 54 (Previously Presented): The computer implemented method according to claim 53, wherein replicating the resources further comprises:
recording, on the set of clones, messages received by the primary node, the messages being injected into the clone elected as the new primary node when switching.

Claim 55 (Previously Presented): The computer implemented method according to claim 40, wherein replicating the resources further comprises:
optimization of information processing resources by load sharing and dynamic process distribution.

Claim 56 (Previously Presented): The computer implemented method according to claim 40, wherein replicating the resources further comprises:

performing non-interruptive maintenance by process relocation upon request, over a data-processing resource network.

Claim 57 (Previously Presented): The computer implemented method according to claim 40, wherein replicating the resources further comprises:

preserving applicative context in a mobile application.

Claim 58 (Currently Amended): A multi-computer system for ensuring functional continuity, capable of running, on at least one computer, at least one software application, the multi-computer system comprising:

a memory comprising a set of instructions;

a processor connected to the memory, capable of executing the set of instructions to implement a method comprising:

ensuring functional continuity of the software application in a multi-computer architecture cluster, the software application being executed at a given time on one of the computers of the cluster, called a primary node, to form a master application, while other computers of the cluster are called a set of secondary nodes, wherein ensuring functional continuity further comprises:

identifying resources and dependencies required by the master application to form required resources;

updating the required resources dynamically on the primary node;

generating a structure of the master application and a dynamic graph of the required resources from the required resources;

replicating the resources by transferring the structure to a set of secondary nodes via a network to form a replica by checkpointing the resources on the set of secondary nodes [[:]], wherein the set of secondary nodes comprises one or more secondary nodes, wherein the replicated resources on the set of secondary nodes comprise a virtual memory and calling stack of each process of the master application, at least one

system resource used by the master application, and data written on disks used by the master application, wherein the checkpointing has an adjustable period to optimize the difference between the recovery time after switching to the set of secondary nodes and the quantity of information to be captured and transferred to the set of secondary nodes;

restoring the replica on the set of secondary nodes to form a set of clone software applications, wherein the set of clone software applications comprises one or more clone software applications;

executing the set of clone software applications on the set of secondary nodes, without loss of context; and

updating the set of clone software applications with incremental updates of the required resources of the master application to create a hot standby application.